

ITWILL 2019년도 7기 자바(JAVA) 개발자 양성 과정

개인 프로젝트_1차

[목차]

String Index;

```
switch(index) {  
    case "3 page":  
        Ch.01 : 프로그램_발전; break;  
    case "7 page":  
        Ch.02 : 프로그램_완성; break;  
    case "9 page":  
        Ch.03 : 구현_입력기능; break;  
    case "14 page":  
        Ch.04 : 구현_검색기능; break;  
    case "19 page":  
        Ch.05 : 구현_수정기능; break;  
    case "22 page":  
        Ch.06 : 구현_삭제기능; break;  
    default:  
        Ch.07 : 감사합니다; break;  
}
```

[기획]

- 다수의 고객에 대한 수월한 관리
- 간편한 데이터 입력, 수정 및 삭제

[사용기술]

- Language : Java
- DB : Oracle 11g

Ch.01 : 프로그램 발전

- ITWILL의 [빅데이터 분석 및 스프링 프레임워크 기반 자바(JAVA) 개발자 양성과정] 수강
- 약 6주간의 교육 후 Swing을 이용한 첫 개인 프로젝트 진행

※ 첫 번째 버전 : 기본 디자인 구상 후 Layout만 구현한 최초 구현 모습입니다.

Ch.01 : 프로그램 발전

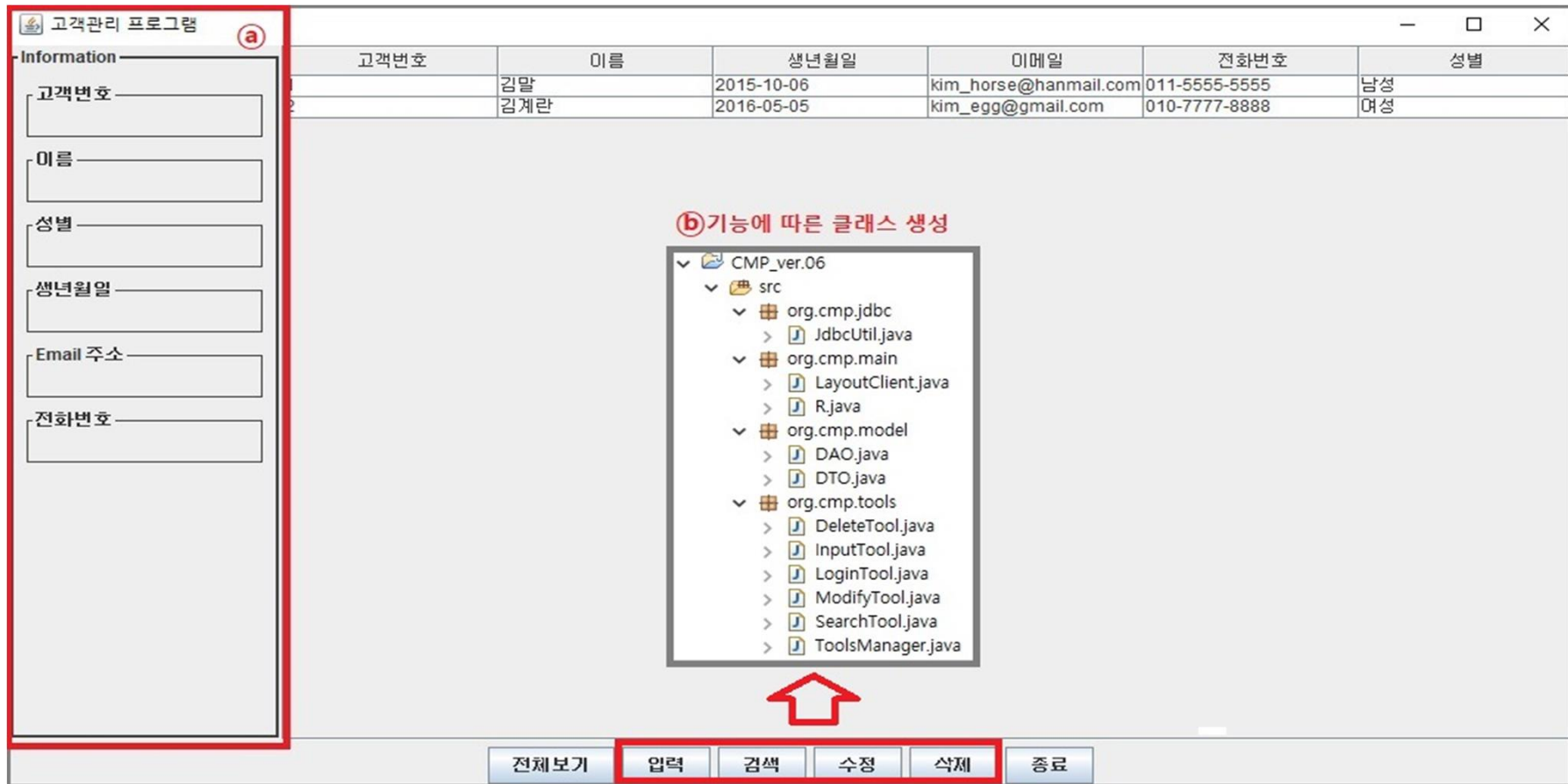
※ 세 번째 버전 : 왼쪽 Information Layout에는 데이터를 순수하게 출력만 하기 위해서 각각의 jTextField에 setEditable 속성을 false로 주었습니다.

고객번호	이름	생년월일	이메일	전화번호	성별
1	김말	2015-10-06	kim_horse@hanmail.c...	011-5555-5555	남성
2	김계란	2016-05-05	kim_egg@gmail.com	010-7777-8888	여성

*setEditable(false) 소스코드 : `tfNo.setEditable(false);`

Ch.01 : 프로그램 발전

- ※ 여섯 번째 버전 : 1. 미적 요소를 위하여 jTextField를 setBorder로 감싸 주었습니다.
2. 여러 가지 기능을 구현하는데 있어 편리함을 얻기 위하여 추가적인 클래스들을 생성하였습니다.



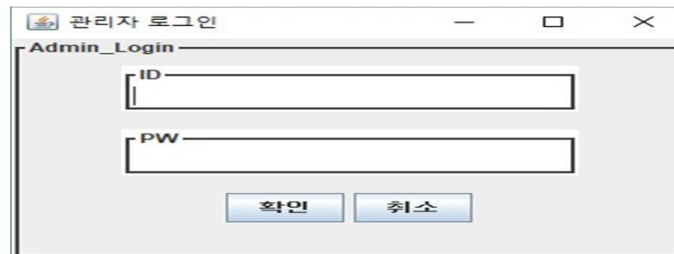
*setBorder 소스코드 : `tfNo.setBorder(new TitledBorder(new LineBorder(Color.DARK_GRAY), "고객번호"));`

Ch.01 : 프로그램 발전

※ 일곱 번째 버전 : 간단한 관리자 로그인 창 구현 모습입니다.

* 관리자 아이디와 비밀번호로만 로그인 할 수 있게 하였습니다.

* 추가적인 기능 추가(㉠Enter key pressed event / ㉡“확인” Button click event)



㉠Enter key pressed event

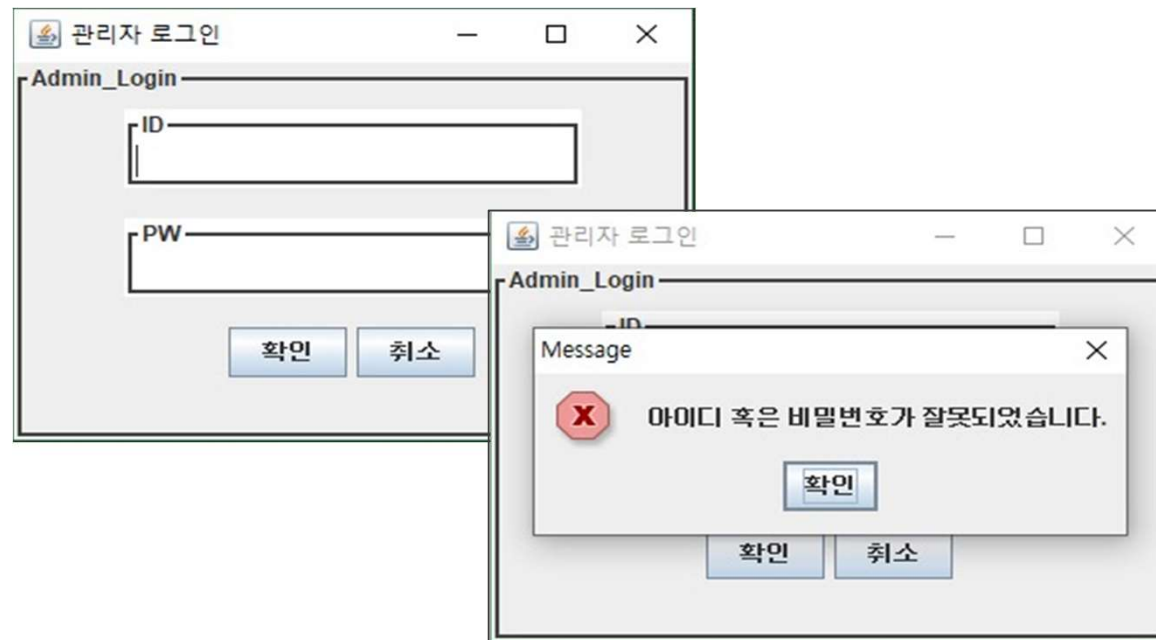
```
tfpw.addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.getKeyCode() == KeyEvent.VK_ENTER) {
            if (tfid.getText().isEmpty() &&
                tfpw.getText().isEmpty()) {
                JOptionPane.showMessageDialog(null,
                    "아이디와 비밀번호를 입력하세요.",
                    "Message", JOptionPane.ERROR_MESSAGE);
            } else if (tfid.getText().equals("comstudy21") &&
                tfpw.getText().contentEquals("comstudy21")) {
                login();
            } else {
                JOptionPane.showMessageDialog(null,
                    "아이디 혹은 비밀번호가 잘못되었습니다.",
                    "Message", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
});
```

㉡“확인” Button click event

```
btnloginOK.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (tfid.getText().isEmpty() &&
            tfpw.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null,
                "아이디와 비밀번호를 입력하세요.",
                "Message", JOptionPane.ERROR_MESSAGE);
        } else if (tfid.getText().equals("comstudy21") &&
            tfpw.getText().contentEquals("comstudy21")) {
            login();
        } else {
            JOptionPane.showMessageDialog(null,
                "아이디 혹은 비밀번호가 잘못되었습니다.",
                "Message", JOptionPane.ERROR_MESSAGE);
        }
    }
});
```

Ch.02 : 프로그램 완성

※ 열 한 번째 버전(프로그램 완성) : 로그인 기능 구현



Ch.02 : 프로그램 완성

※ 열 한 번째 버전(프로그램 완성) : 고객관리 프로그램 실행 화면

- * 좌측 Information에 이미지 파일 출력 기능을 추가하였습니다(입력 기능 설명에 더 자세히...).
- * 프레임 타이틀에 '오늘날짜'를 추가하였습니다.

◆고객관리 프로그램◆ 날짜 : 2019년/10월/20일

이름	이메일	전화번호	성별

Information

사진

고객번호

이름

성별

생년월일

Email 주소

전화번호

초기화

©버찌 corp.

전체보기 입력 검색 수정 삭제 종료

※"오늘날짜" 소스코드 :

```
private String inDate = new java.text.SimpleDateFormat
("yyyy년/MM월/dd일").format(new java.util.Date());
```


Ch.03 : 구현 입력기능

※ **입력기능** : 고객관리 프로그램에서 "입력" 버튼을 클릭하면 **새로운 창이 생성** 됩니다.

* 과정 : 새로운 창 생성 -> [㉠사용 된 기술]을 통해 문자열로 데이터를 입력 받음 -> 입력 받은 데이터를 [㉢DAO에서 처리]

[㉠사용 된 기술]

1. JTextField
2. JRadioButton
3. JComboBox 등

[㉢DAO에서 처리]

```
public void insert(DTO dto) {  
    try {  
        pstmt = conn.prepareStatement(INSERT);  
        pstmt.setString(1, dto.getcName());  
        pstmt.setString(2, dto.getcBirth());  
        pstmt.setString(3, dto.getcMail());  
        pstmt.setString(4, dto.getcTel());  
        pstmt.setString(5, dto.getcGender());  
        pstmt.setString(6, dto.getcImgPath());  
        int cnt = pstmt.executeUpdate();  
        if(cnt>0) {  
            System.out.println("::: 입력 완료 :::");  
            conn.commit();  
        } else {  
            System.out.println("::: 입력 실패 :::");  
            conn.rollback();  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        JdbcUtil.close(pstmt);  
    }  
}  
  
} // end method insert
```

```
INSERT = "INSERT INTO CUSTOMER VALUES (cmadmin_seq.nextval,?, ?, ?, ?, ?, ?)";
```

Ch.03 : 구현 입력기능

※ 입력기능 : 생년월일 입력

입력

Input

고객이름 김말

생년월일 2019 년 01 월 01 일

전자우편 2019 @ gmail.com

전화번호 2012

사진위치

남성 여성

확인 닫기

입력

Input

고객이름 김말

생년월일 2015 년 01 월 01 일

전자우편 008 il.com

전화번호 010 12

사진위치

남성 여성

확인 닫기

입력

Input

고객이름 김말

생년월일 2015 년 10 월 01 일

전자우편 004 gmail.co

전화번호 010 08

사진위치

남성 여성

확인 닫기

Ch.03 : 구현 입력기능

※ 입력기능 : 이메일 주소와 전화번호 입력

The screenshot shows the '입력' (Input) window with the following fields and values:

- 고객이름: 김말
- 생년월일: 2015 년 10 월 06 일
- 전자우편: kim_horse @ gmail.com (dropdown menu is open showing options: gmail.com, naver.com, nate.com, hanmail.com, hotmail.com, outlook.kr)
- 전화번호: 010 (dropdown menu is open showing options: 010, 011, 016, 017)
- 사진위치: (empty)
- 성별: 남성 (selected)

Buttons: 검색, 확인, 닫기

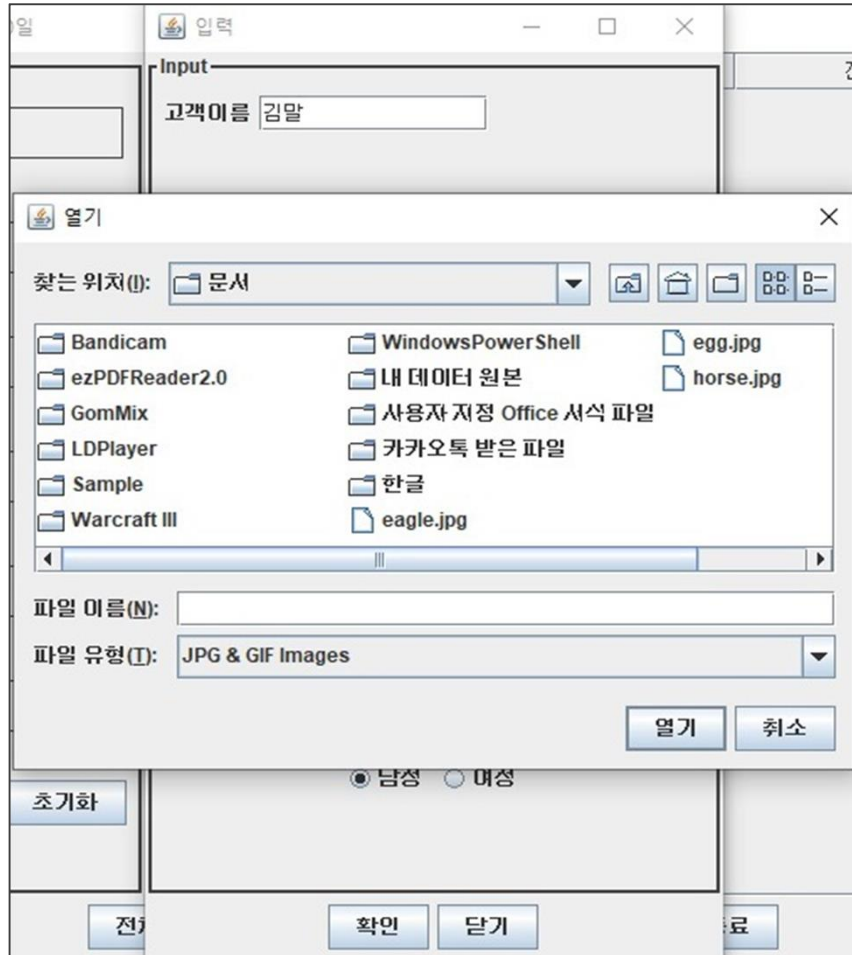
The screenshot shows the '입력' (Input) window with the following fields and values:

- 고객이름: 김말
- 생년월일: 2015 년 10 월 06 일
- 전자우편: kim_horse @ hanmail.com
- 전화번호: 010 (dropdown menu is open showing options: 010, 011, 016, 017)
- 사진위치: (empty)
- 성별: 남성 (selected)

Buttons: 검색, 확인, 닫기

Ch.03 : 구현 입력기능

※ 입력기능 : 사진 입력기능과 소스코드



```
itbtnImginput = new JButton("검색");
ittfImgPath = new JTextField(18);
itbtnImginput.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JFileChooser chooser = new JFileChooser();
        FileNameExtensionFilter filter =
            new FileNameExtensionFilter("JPG & GIF Images",
                "jpg", "gif");
        chooser.setFileFilter(filter);

        int ret = chooser.showOpenDialog(null);
        if(ret != JFileChooser.APPROVE_OPTION) {
            JOptionPane.showMessageDialog(null,
                "파일을 선택하지 않았습니다", "경고",
                JOptionPane.WARNING_MESSAGE);
            return;
        }
        itfilePath = chooser.getSelectedFile().getPath();
        ittfImgPath.setText(itfilePath);
        return;
    }
});
```



Ch.03 : 구현 입력기능

※ 입력기능 : 입력완료 후 우측의 입력 된 데이터를 클릭 시 좌측 information에 출력

◆고객관리 프로그램◆ 날짜 : 2019년/10월/20일

Information

사진



고객번호

1

이름

김말

성별

남성

생년월일

2015-10-06

Email 주소

kim_horse@hanmail.com

전화번호

011-5555-5555

초기화

이름

김말

전체보기 입력 검색

```
@Override
public void mousePressed(MouseEvent e) {
    int row = table.getSelectedRow();
    String pCode = model.getValueAt(row, 0).toString();
    String pName = model.getValueAt(row, 1).toString();
    String pGender = model.getValueAt(row, 5).toString();
    String pBirth = model.getValueAt(row, 2).toString();
    String pEmail = model.getValueAt(row, 3).toString();
    String pPhone = model.getValueAt(row, 4).toString();
    dto.setcImgPath(model.getValueAt(row, 6).toString());

    tfNo.setText(pCode);
    tfName.setText(pName);
    tfGender.setText(pGender);
    tfBirth.setText(pBirth);
    tfEmail.setText(pEmail);
    tfPhone.setText(pPhone);
    imgLb.setIcon(new ImageIcon(
        new ImageIcon(dto.getcImgPath()).getImage().
        getScaledInstance(160, 200, Image.SCALE_DEFAULT)));
}
```


Ch.04 : 구현 검색기능

※ 검색기능 : [검색]버튼을 클릭하면 새로운 창이 생성되며 검색 기능을 수행합니다.

[DTO]

```
// 데이터 검색용
public Vector<String> getList() {
    Vector<String> list = new Vector<>();
    list.add(cCode+"");
    list.add(cName);
    list.add(cBirth);
    list.add(cMail);
    list.add(cTel);
    list.add(cGender);
    list.add(cImgPath);
    return list;
}
```

[DAO]

```
public Vector<Vector> selectOne(DTO dto) {
    Vector<Vector> searchList = new Vector<>();
    // create -> set -> execute 순서 항상 알기!!!
    // 만들고 -> 사용하고 -> 받아오기
    try {
        pstmt = conn.prepareStatement(SELECT_CODE);
        pstmt.setInt(1, dto.getcCode());
        rs = pstmt.executeQuery();
        while (rs.next()) {
            dto.setcCode(rs.getInt(1));
            dto.setcName(rs.getString(2));
            dto.setcBirth(rs.getString(3));
            dto.setcMail(rs.getString(4));
            dto.setcTel(rs.getString(5));
            dto.setcGender(rs.getString(6));
            dto.setcImgPath(rs.getString(7));

            searchList.add(dto.getList());
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtil.close(null, pstmt, rs);
    }
    stData = searchList;
    return stData;
} // end method selectOne
```

```
SELECT_CODE = "SELECT * FROM CUSTOMER WHERE CCODE=?";
```

Ch.04 : 구현 검색기능

※ 검색기능 : 검색한 데이터에 대한 유효성 검사

```
// 유효성 검사
public void validation () {
    if(sttfNo.getText().isEmpty()) {
        JOptionPane.showMessageDialog(null, "고객번호를 입력하세요.",
            "Message", JOptionPane.ERROR_MESSAGE);
    } else {
        dto.setcCode(new Integer(sttfNo.getText()));
        // 데이터 존재 유무 판단용
        dao.selectOne(dto);
        // 검색한 데이터가 없을 때 경고창 띄우기
        if(stData.isEmpty()) {
            JOptionPane.showMessageDialog(null, "해당 데이터가 없습니다.",
                "Message", JOptionPane.ERROR_MESSAGE);
            sttfNo.setText("");
            ToolsManager.changeTableData();
        } else {
            // static에다가 넣어줘야 함
            System.out.println("### test ###" + stData);

            columnNames = new Vector<>();
            for (String column : txtArr) {
                columnNames.add(column);
            }
            // 화면에 검색 된 데이터 출력
            model.setDataVector(stData, columnNames);
        }
    }
}
```

Ch.04 : 구현 검색기능

※ 검색기능 : 프로그램 상단에 검색 창 생성

검색

Search

고객번호

확인

◆고객관리 프로그램◆ 날짜 : 2019년/10월/20일

Information

사진

고객번호

이름

성별

생년월일

Email 주소

전화번호

초기화

이름	이메일	전화번호	성별
김말	kim_horse@hanmail.com	011-5555-5555	남성
김계란	kim_egg@gmail.com	010-7777-8888	여성
김독수리	kim_eagle@outlook.kr	017-7080-7080	남성

©버찌 corp.

전체보기

입력

검색

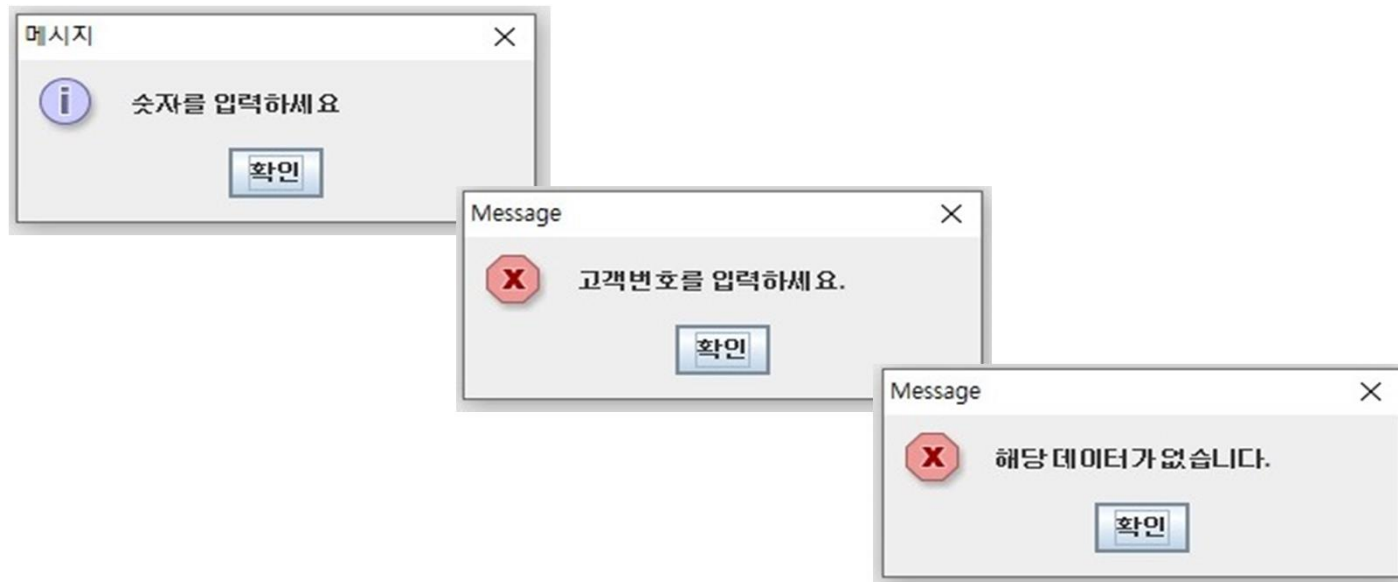
수정

삭제

종료

Ch.04 : 구현 검색기능

※ 검색기능 : 유효성 검사 결과에 따른 메시지



Ch.04 : 구현 검색기능

※ 검색기능 : 검색 완료

고객관리 프로그램 날짜 : 2019년/10월/20일

Information

사진

고객번호 3

이름 김독수리

성별 남성

생년월일 1960-09-03

Email 주소 kim_eagle@outlook.kr

전화번호 017-7080-7080

초기화

이름	이메일
김독수리	kim_eagle@outlook.kr

전체보기 입력 검색 수정 삭제

Ch.05 : 구현 수정기능

※ 수정기능 : 수정 기능은 (검색 기능 + 입력 기능)과 비슷합니다.

기존에 만들어놓은 검색 기능을 활용하여
[고객번호]로 수정하고자 하는 데이터를 가져옵니다.



고객번호를 입력하면 그에 해당하는 데이터를 가져와서 각각의 컴포
넌트에 입력하고, 고객번호를 제외한 나머지 모든 컴포넌트들이
setEditable(true)로 변하게 합니다.

Ch.05 : 구현 수정기능

※ 수정기능 : 전화번호나 생년월일 같이 DB에 하나의 문자열로 저장된 데이터를 StringTokenizer를 활용해서 나누어 받을 수 있었습니다.

[StringTokenizer] 소스코드

```
// StringTokenizer로 birth 데이터를 분해해서 세 개로 나눈 다음에 각각 따로따로 넣어 줌
StringTokenizer st1 = new StringTokenizer(cBirth, "-");
String st1Year = st1.nextToken();
String st1Month = st1.nextToken();
String st1Day = st1.nextToken();
myearCombo.setSelectedItem(st1Year);
mtmonthCombo.setSelectedItem(st1Month);
mtdayCombo.setSelectedItem(st1Day);
```

```
// 이메일 나오게 하기
StringTokenizer st2 = new StringTokenizer(cEmail, "@");
String st2Id = st2.nextToken();
String st2Adress = st2.nextToken();
mttfEmail.setText(st2Id);
mtemailCombo.setSelectedItem(st2Adress);

// 전화번호 나오게 하기
StringTokenizer st3 = new StringTokenizer(cPhone, "-");
String st3Phone1 = st3.nextToken();
String st3Phone2 = st3.nextToken();
String st3Phone3 = st3.nextToken();
phoneCombo.setSelectedItem(st3Phone1);
mttfPhoneMid.setText(st3Phone2);
mttfPhoneEnd.setText(st3Phone3);
```

Ch.05 : 구현 수정기능

※ 수정기능 : DAO에서의 데이터 처리

* executeUpdate가 실행 되면 숫자 1을 반환하기 때문에 이를 활용하여 간단하게 조건을 생성하였습니다.

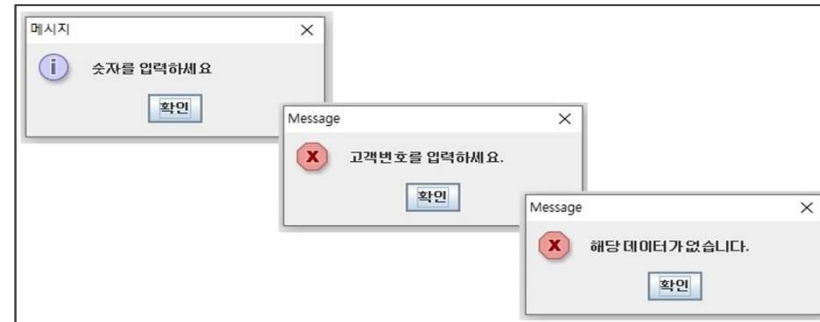
```
public void update(DTO dto) {
    try {
        // sql문 보내기
        pstmt = conn.prepareStatement(UPDATE);

        // 수정할 내용 입력
        pstmt.setString(1, updateDTO.getcName());
        pstmt.setString(2, updateDTO.getcBirth());
        pstmt.setString(3, updateDTO.getcMail());
        pstmt.setString(4, updateDTO.getcTel());
        pstmt.setString(5, updateDTO.getcGender());
        pstmt.setString(6, updateDTO.getcImgPath());
        // 조건 입력
        pstmt.setInt(7, updateDTO.getcCode());
        if(pstmt.executeUpdate() > 0) {
            JOptionPane.showMessageDialog(null, "수정 완료",
                "Message", JOptionPane.INFORMATION_MESSAGE);
            conn.commit();
        } else {
            JOptionPane.showMessageDialog(null, "수정 불가능",
                "Message", JOptionPane.ERROR_MESSAGE);
            conn.rollback();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtil.close(pstmt);
    }
    // 수정 완료 후 수정할 때 입력 한 고객번호를 빈공간으로 초기화
    mttfNo.setText("");
} // end method update
```

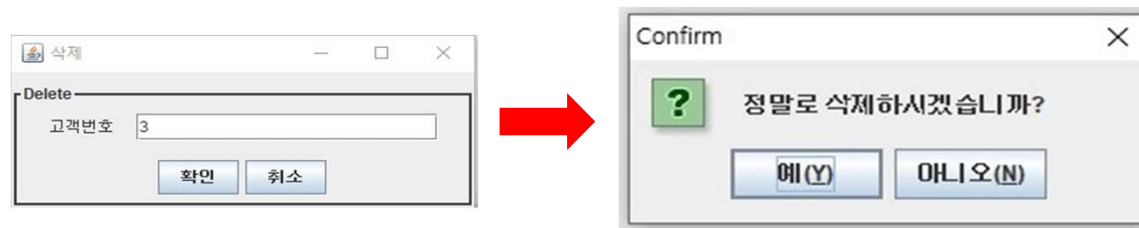
```
UPDATE = "UPDATE CUSTOMER SET CNAME=?,CBIRTH=?,CMAIL=?,CTEL=?,CGENDER=?,CIMGPATH=? WHERE CCODE=?";
```

Ch.06 : 구현 삭제기능

※ 삭제기능 : 삭제 시도 시 고객번호를 입력 받고, 이에 대한 유효성 검사는 검색 기능에 있는 것을 재활용하였습니다.



삭제 기능에는 데이터를 삭제 여부를
다시 한 번 물어보는 기능을 추가하였습니다.



Ch.06 : 구현 삭제기능

※ 삭제기능 : 유효성 검사와 DAO에서의 데이터 처리

```
// 유효성 검사
public void validation() {
    if (dttfNo.getText().isEmpty()) {
        JOptionPane.showMessageDialog(
            null, "고객번호를 입력하세요",
            "Message", JOptionPane.ERROR_MESSAGE);
    } else {
        dto.setCcode(new Integer(dttfNo.getText()));
        // 데이터가 있는지 없는지 판단하기 위해 selectOne에 데이터를 같이 보냄
        dao.selectOne(dto);
        // 검색한 데이터가 없을 때 경고창 띄우기
        if (stData.isEmpty()) {
            JOptionPane.showMessageDialog(
                null, "해당 데이터가 없습니다.",
                "Message", JOptionPane.ERROR_MESSAGE);
            dttfNo.setText("");
            ToolsManager.changeTableData();
        } else {
            int question = JOptionPane.showConfirmDialog(
                null, "정말로 삭제하시겠습니까?",
                "Confirm", JOptionPane.YES_NO_OPTION);
            dttfNo.setText("");
            if (question == JOptionPane.YES_OPTION) {
                dao.delete(dto);
                dttfNo.setText("");
                ToolsManager.changeTableData();
            } else {
                dttfNo.setText("");
            }
        }
    }
}
```

```
public void delete(DTO dto) {
    /*DELETE FROM CUSTOMER WHERE CCODE=?*/
    try {
        pstmt = conn.prepareStatement(DELETE);
        pstmt.setInt(1, dto.getCcode());
        if(0 < pstmt.executeUpdate()) {
            System.out.println("### test ### 삭제 완료");
            conn.commit();
        } else {
            System.out.println("### test ### 삭제 안 됨");
            conn.rollback();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        JdbcUtil.close(pstmt);
    }
} // end method delete
```

```
DELETE = "DELETE FROM CUSTOMER WHERE CCODE=?";
```

Ch.07 : 감사합니다.

프로젝트를 마무리하며...

대학교에서 피아노를 전공하고 돌연 조교로 취업을 해서 일을 하던 중 우연치 않게 발견한 개발자라는 직업에 도전 할 엄두도 내지 못할 때, ITWILL의 국비과정을 통해 개발자로 조금씩 성장할 수 있었습니다.

물론 비전공자로서 미래에 대한 불확실성 때문에 늘 불안함 가운데 지냈지만 이런 저에게 매번 큰 위로와 희망을 주시는 좋으신 선생님을 만나 부정적인 마음가짐을 극복해낼 수 있었습니다.

이제 겨우 프로젝트 하나를 마무리했다고 생각했지만 다시 생각하면 약 7주 만에 어려운 Java Swing 프로젝트를 마무리 했다는 사실에 가슴이 벅칩니다.

처음에는 코딩의 편의를 위해 Class를 여러 개 생성하는 것이 적응되지 않았고 또 어려웠지만 불과 얼마 전, 효율적인 코딩을 위해 새로운 Class를 자연스럽게 추가하고 있는 제 모습에 웃음이 나왔습니다.

마지막으로 Oracle DB에 테이블을 생성하고 Java에서 SQL문을 이용하는 등 Oracle을 다뤄본 것, Java Swing의 다양한 컴포넌트들을 활용하여 기초적인 CRUD 기능과 UI를 구현하고 이벤트 리스너를 다루는 것 등의 경험을 통해 코딩에 자신감을 얻을 수 있었습니다.

감사합니다.

ITWILL 빅데이터 분석 및 스프링 프레임워크 기반 자바(JAVA) 개발자 양성과정
2019년도 제 7기 강 한결 드림